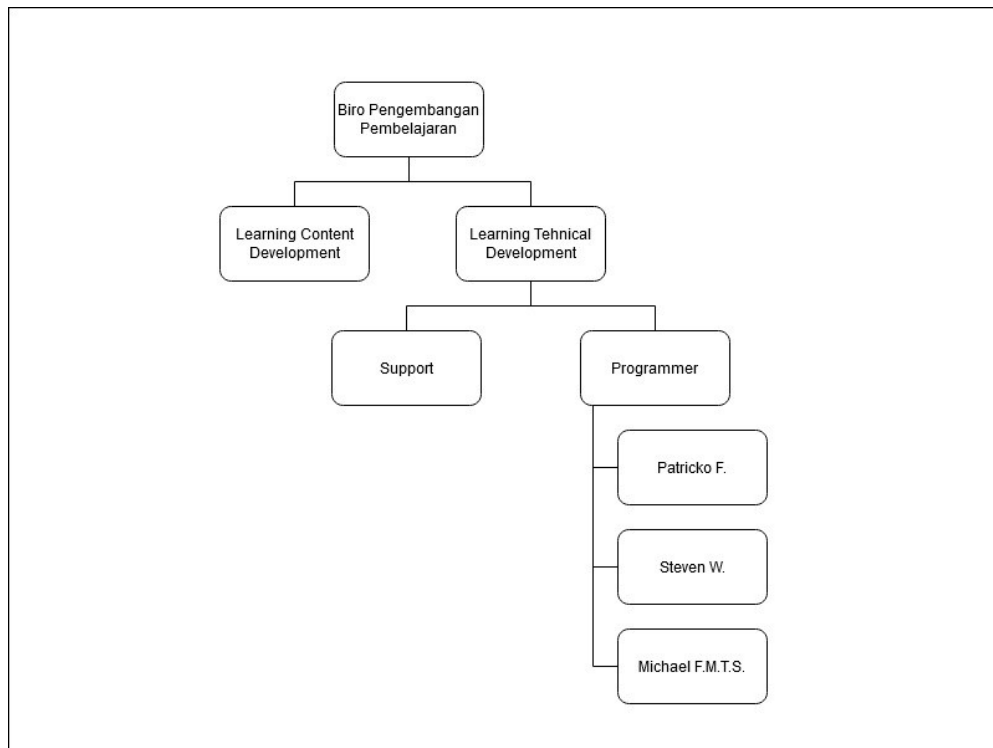


BAB 3

PELAKSANAAN KERJA MAGANG

3.1. Kedudukan dan Koordinasi

Struktur kedudukan dalam organisasi selama melakukan kerja magang di Biro Pengembangan Pembelajaran (BPP) adalah sebagai berikut.



Gambar 3.1 Stuktur Organisasi Internal Biro Pengembangan Pembelajaran

Kerja magang sebagai *programmer* di BPP dibimbing oleh Christian Orvin selaku *supervisor* magang yang menjabat sebagai Koordinator divisi Learning Technical Development. *Supervisor* memberikan kriteria-kriteria dari sistem yang dibutuhkan dalam pengembangan Learning Center Portal. Kemudian perancangan sistemnya dilakukan secara bersama-sama.

Pengembangan sistem *Learning Center Portal* dikerjakan oleh pelaksana kerja magang yang menjabat sebagai programmer. Sistem dibuat dengan bantuan dan bimbingan dari programmer senior dan *user interface / user experience (UI/UX) designer*. Tampilan dan sistem dari *Learning Center Portal* didesain mengikuti arahan dari *supervisor*. Yang jika ada kesalahan atau tidak sesuai dari sistemnya maka akan dilakukan perubahan atau perbaikan secara langsung. Kendala-kendala lain yang muncul akan segera didiskusikan juga dalam satu tim.

3.2. Tugas yang Dilakukan

Berikut adalah tugas yang dilakukan selama melakukan kerja magang di Biro Pengembangan Pembelajaran.

1. Membuat website *Learning Center Portal* dengan menggunakan *framework* Laravel.
2. Menggabungkan monitoring, helpcenter, dan request management system ke Learning Center Portal.
3. Membuat fitur untuk melakukan filter.
4. Membuat fitur request kebutuhan.
5. CMS untuk tukar PIC.
6. Membuat kelas zoom untuk minggu pertama.
7. Membuat CMS untuk admin.

Berikut adalah *timeline* kerja magang yang dilaksanakan dalam pengerjaan penambahan fitur pada LCP.

Nama kegiatan	Minggu								
	1	2	3	4	5	6	7	8	9
Mempelajari <i>framework</i> Laravel									
Pembuatan dan perubahan pada halaman utama LCP									
Pembuatan request kebutuhan									
Pengubahan semua halaman jadi <i>import</i> aset dari lokal									
Pembuatan filter									
Pembuatan CMS sistem pertukaran PIC									
Pembuatan CMS									

Tabel 3.1 Timeline kerja magang

3.3. Uraian Pelaksanaan Kerja Magang

Berikut adalah beberapa *software* yang digunakan dalam pembangunan Learning Center Portal.

1. WinSCP

WinSCP merupakan aplikasi yang digunakan untuk melakukan *File Transfer Protocol* (FTP). Aplikasi ini digunakan untuk melakukan FTP ke server development E-Learning UMN.

2. Visual Studio Code

Visual Studio Code merupakan aplikasi text editor yang berguna untuk menulis kode dalam pembuatan web. Aplikasi ini mempermudah penulisan kode karena adanya *extension* yang membantu dalam pembuatan web.

3. Notepad++

Notepad++ merupakan text editor juga. Aplikasi ini digunakan juga karena *auto open file* php yang ada di WinSCP menggunakan notepad.

Uraian dari kerja magang di Biro Pengembangan Pembelajaran dibagi menjadi 3 bagian, yaitu perancangan aplikasi, desain tampilan, hasil implementasi halaman, kendala yang ditemukan, dan solusi atas kendala yang ditemukan.

3.3.1. Perancangan Sistem

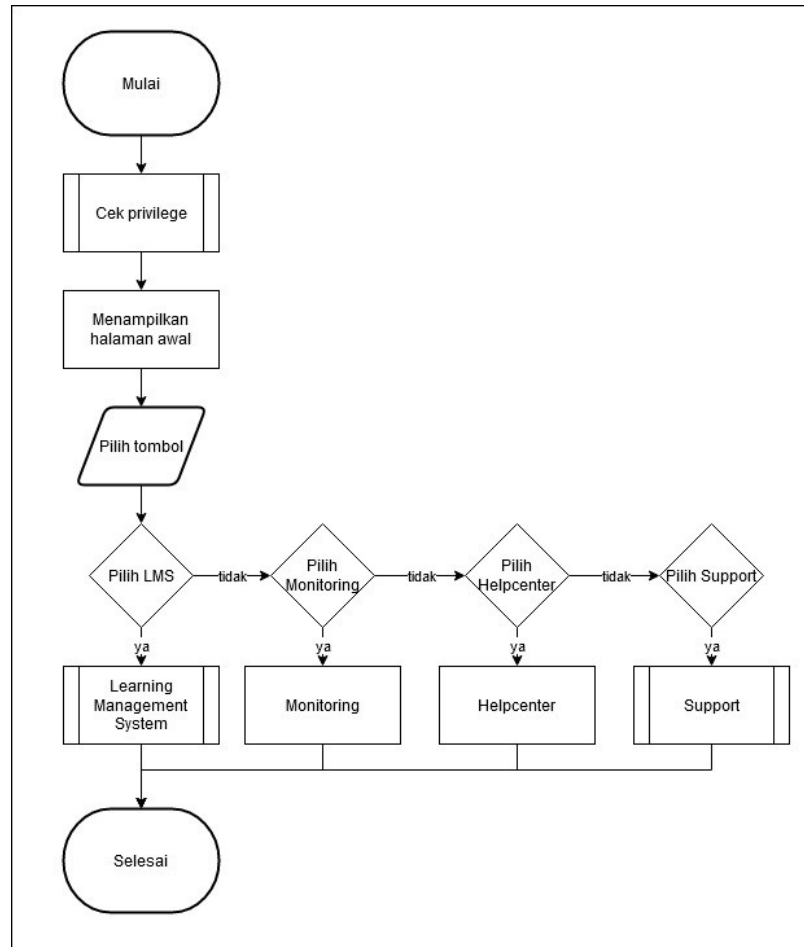
Web dari *Learning Center Portal* (LCP) dibuat dengan menggunakan *framework* Laravel dan databasenya adalah MySQL. Rancangan dari aplikasinya dibuat berdasarkan kebutuhan dari keadaan yang diberitahukan. Berikut adalah *flowchart*, diagram aliran data, dan *class diagram* dari pembuatan system LCP.

A. Flowchart

Berikut adalah *flowchart* dari LCP, fitur filter, *request* kebutuhan, CMS tukar PIC, dan CMS untuk admin.

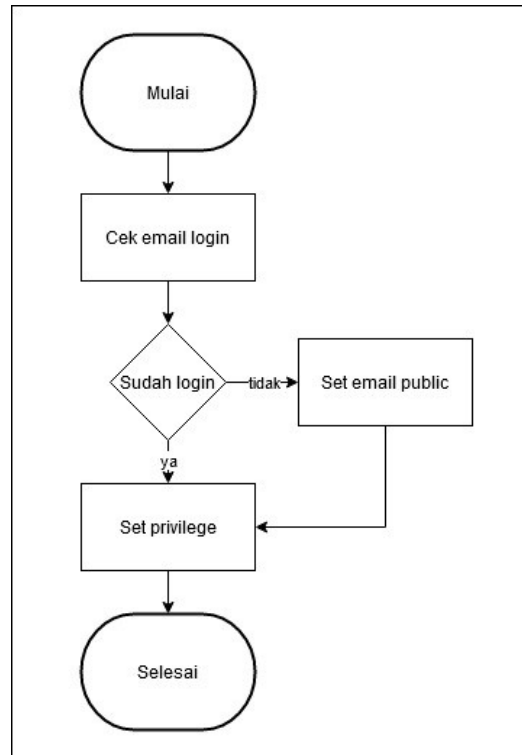
A.1. Learning Center Portal

Halaman LCP yang dibuat terdiri dari 3 bagian, yaitu: halaman menu awal, *learning management system* (LMS), dan *support*. Pada halaman menu awal memiliki 4 tombol, yaitu tombol LMS, monitoring, helpcenter, dan *support*. Masing-masing tombol akan mengunjungi halaman yang diinginkan jika ditekan oleh pengguna. Berikut adalah *flowchart* dari tampilan menu awal.



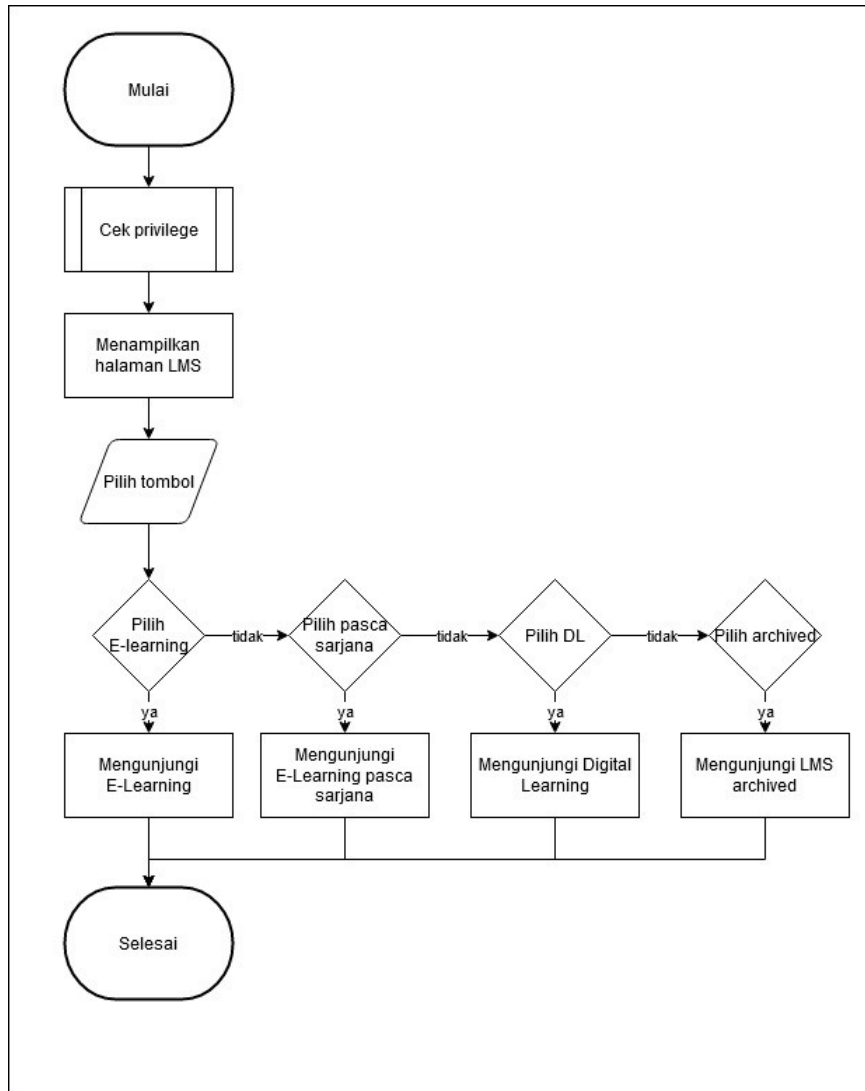
Gambar 3.2 Flowchart menu awal

Berikut adalah *flowchart* dari proses cek *privilege*. Di tahap ini sistem akan mengecek apakah *user* sudah login atau belum. Jika belum login, maka email akan diatur menjadi *public*. Dikarenakan tabel *privilege* yang kredensial, maka pelaksana kerja magang tidak diberikan akses untuk memakainya. Maka dari itu dibuatlah cek *privilege dummy* yang mengecek email dari pengguna yang dilakukan dengan *hardcode* pada *controller* untuk inisialisasi emailnya.



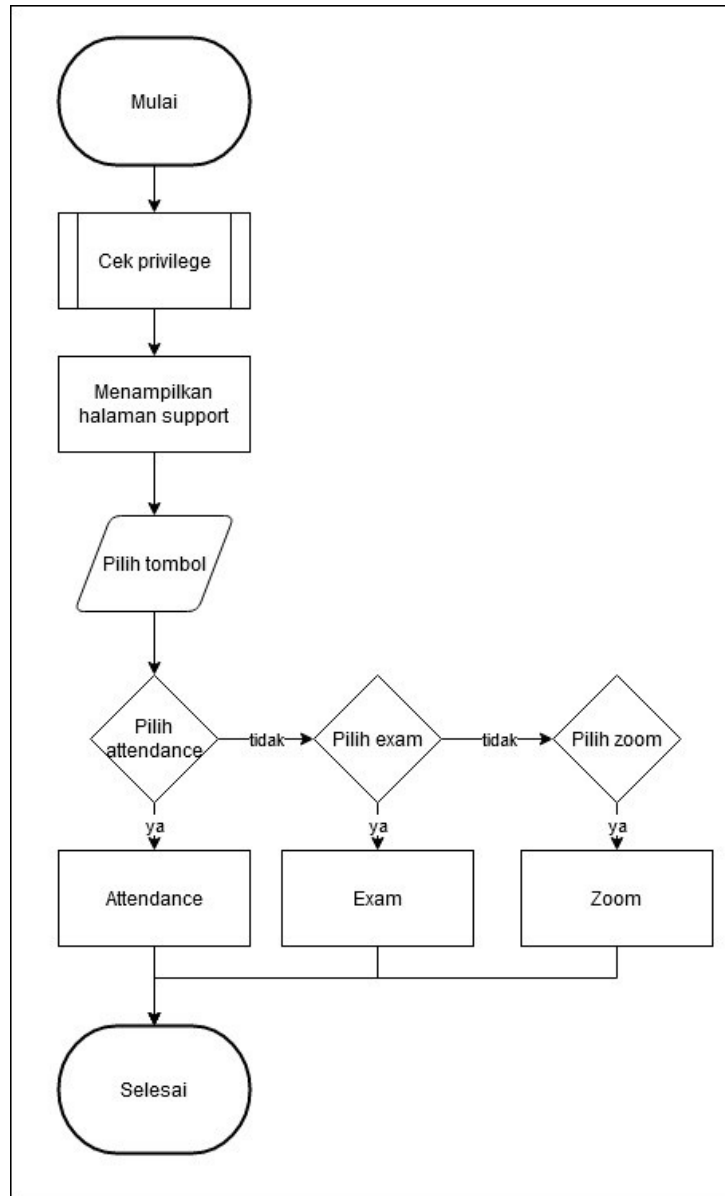
Gambar 3.3 Flowchart cek privilege

Berikut adalah *flowchart* dari menu Learning Management System (LMS). Pada halaman LMS tersedia 4 tombol, yaitu tombol E-Learning dari S1/D3, E-Learning pasca sarjana, digital learning, dan archived. Saat tombol E-Learning S1/D3, E-Learning pasca sarjana, dan digital learning dipilih, *user* akan membuka tab baru untuk mengaksesnya. Sedangkan jika tombol archived dipilih, maka akan membuka halaman yang berisi tentang E-Learning yang terdahulu.



Gambar 3.4 Flowchart menu LMS

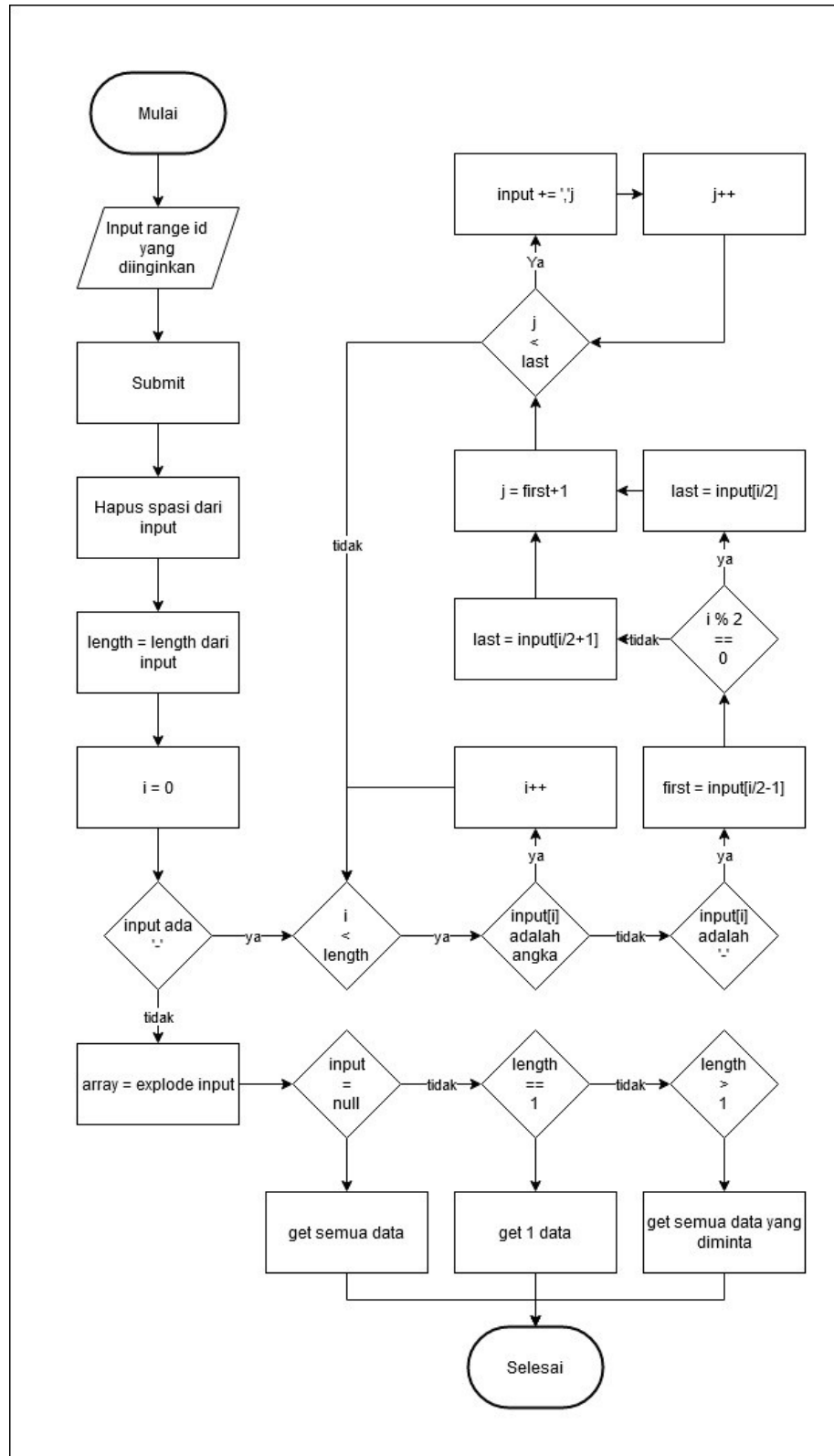
Berikut adalah *flowchart* dari menu support. Pada halaman support akan ada 3 tombol, yaitu attendance, exam, dan zoom. Saat tombol attendance, exam dan zoom diklik, maka akan dialihkan ke halamannya. Perlu diketahui bahwa pelaksana kerja magang hanya menyiapkan tampilan dan *route* untuk ke halaman yang akan dituju sehingga tidak mengetahui bagaimana proses selanjutnya dari masing-masing menu.



Gambar 3.5 Flowchart menu support

A.2. Fitur Filter

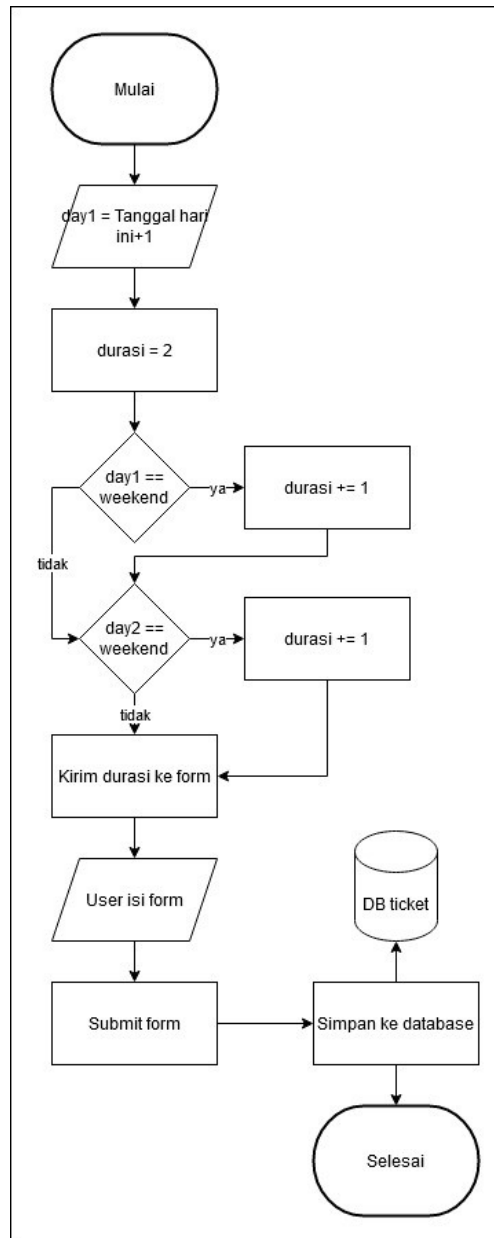
Fungsi filter digunakan untuk memfilter akun-akun LCD yang ingin ditampilkan oleh user. User akan memasukkan ke *form* input akun LCD yang diinginkan, dan listnya akan tertampil setelah tombol submit ditekan. Berikut adalah *flowchart backend* dari bagaimana proses mengolah input user.



Gambar 3.6 Flowchart filter

A.3. Request Kebutuhan

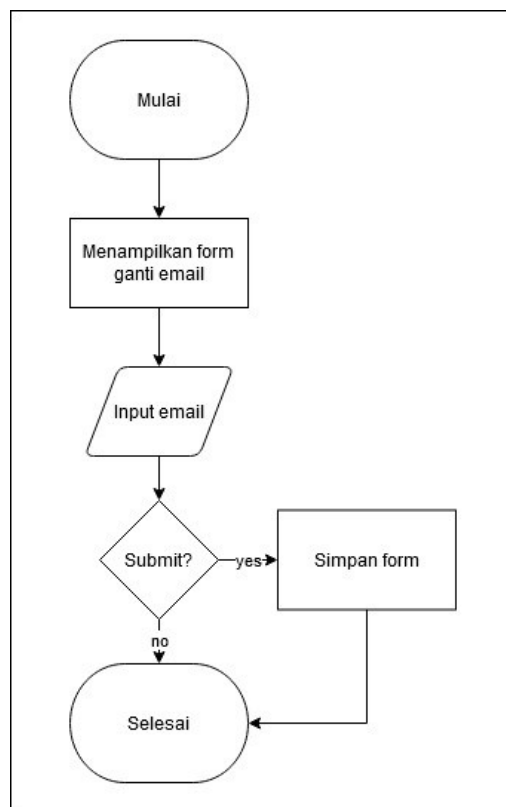
Berikut adalah *flowchart* dari halaman request kebutuhan. Halaman request kebutuhan digunakan oleh *user* untuk memberikan request ke BPP. Durasi dari penyelesaian request dapat diisi dengan minimal 2 hari masa kerja.



Gambar 3.7 Flowchart request kebutuhan

A.4. CMS Tukar PIC

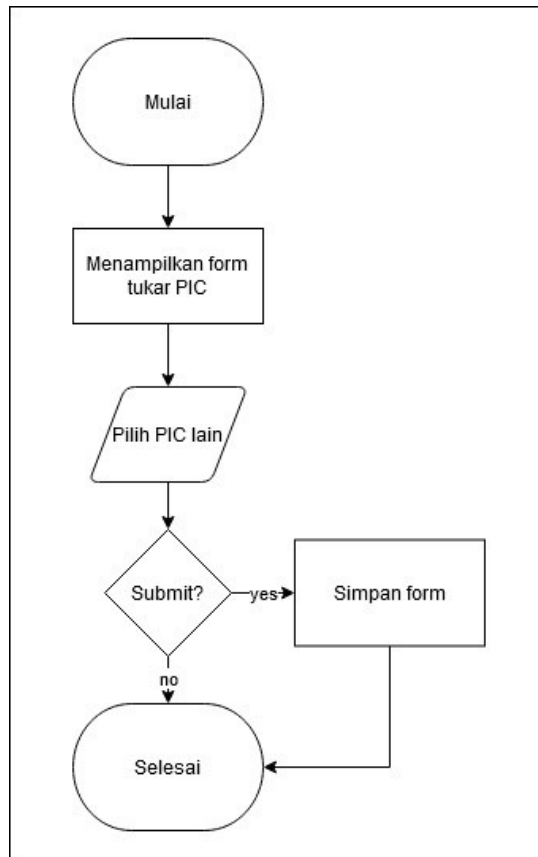
Yang dilakukan sebelum melakukan proses tukar PIC adalah mendaftarkan email dari admin yang akan menjadi PIC. Pendaftaran dilakukan dengan mengisi email pada form sesuai dengan jabatan yang dimiliki. Berikut adalah *flowchart* dari halaman untuk pendaftaran atau penggantian email.



Gambar 3.8 Flowchart ganti email

Setelah pendaftaran email, maka email dari yang bersangkutan akan dapat mengajukan request tukar PIC ke PIC lain yang telah terdaftar. Jika sudah mengirimkan *form* nya maka *form* akan disimpan pada database yang nantinya akan

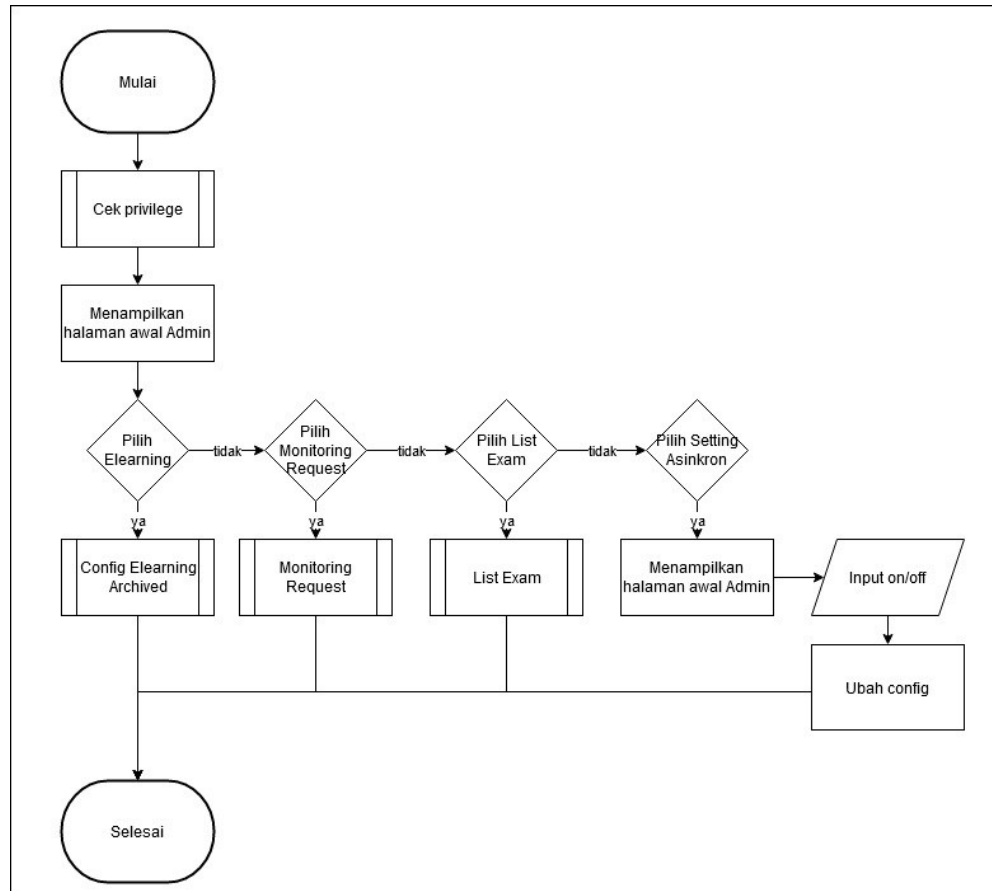
dapat diakses oleh PIC lain untuk disetujui atau ditolak. Berikut adalah *flowchart* dari halaman untuk *request* tukar PIC.



Gambar 3.9 Flowchart request tukar PIC

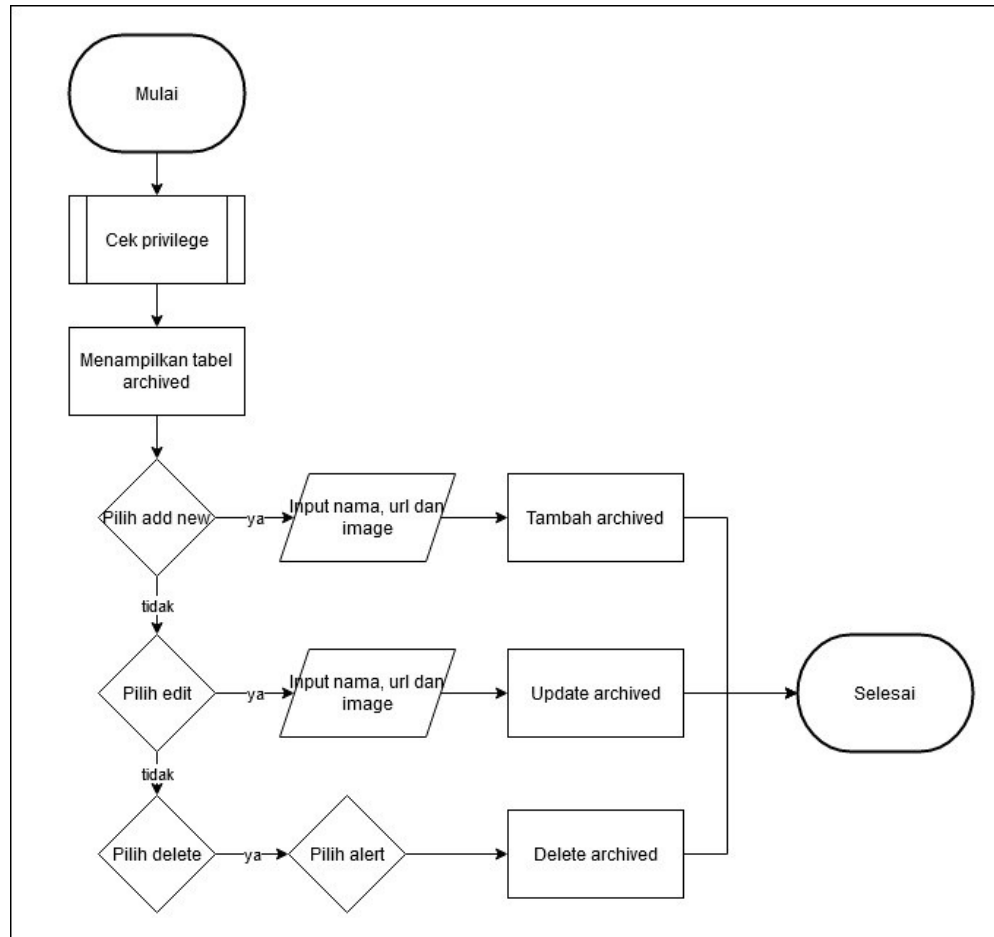
A.5. CMS Admin

Halaman menu awal akan menampilkan daftar-daftar dari menu yang dapat diatur oleh adminnya. Pada saat ini sudah tersedia 4 CMS untuk admin yaitu: pada E-Learning archived, monitoring *request* kebutuhan, pengaturan *list exam*, dan pengaturan asinkron. Jika memilih untuk mengatur asinkron, maka akan ditampilkan *modal* yang dapat mengaktifkan maupun mematikan pengajuan kelas asinkron. Berikut adalah *flowchart* dari halaman menu dari admin.



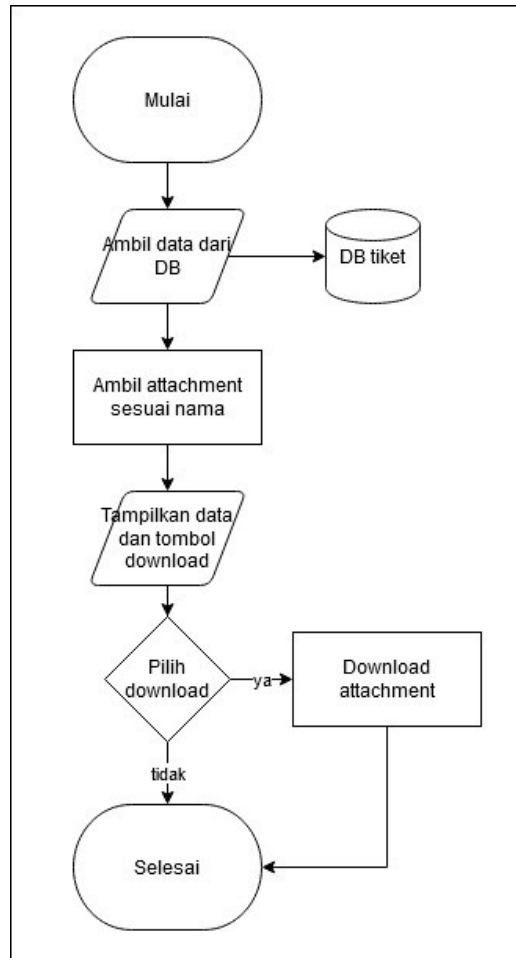
Gambar 3.10 Flowchart menu admin

Pada halaman CMS untuk konfigurasi E-Learning *archived* akan ditampilkan daftar dari E-Learning lama yang sudah didaftarkan. Pada masing-masingnya dapat dilakukan perubahan seperti *update* URL nya, maupun mengganti gambarnya. Penambahan *archived* juga dapat dilakukan dengan menekan tombol ‘*add*’ lalu mengisi formnya. Penghapusan daftar juga dapat dilakukan dengan menekan tombol *delete*. Saat tombol ditekan, sistem akan menampilkan *alert* yang menanyakan apakah yakin untuk menghapusnya. Jika ditekan ‘*ya*’, maka E-Learning tersebut akan terhapus dari daftar. Berikut adalah *flowchart* dari halaman untuk konfigurasi E-Learning *archived*.



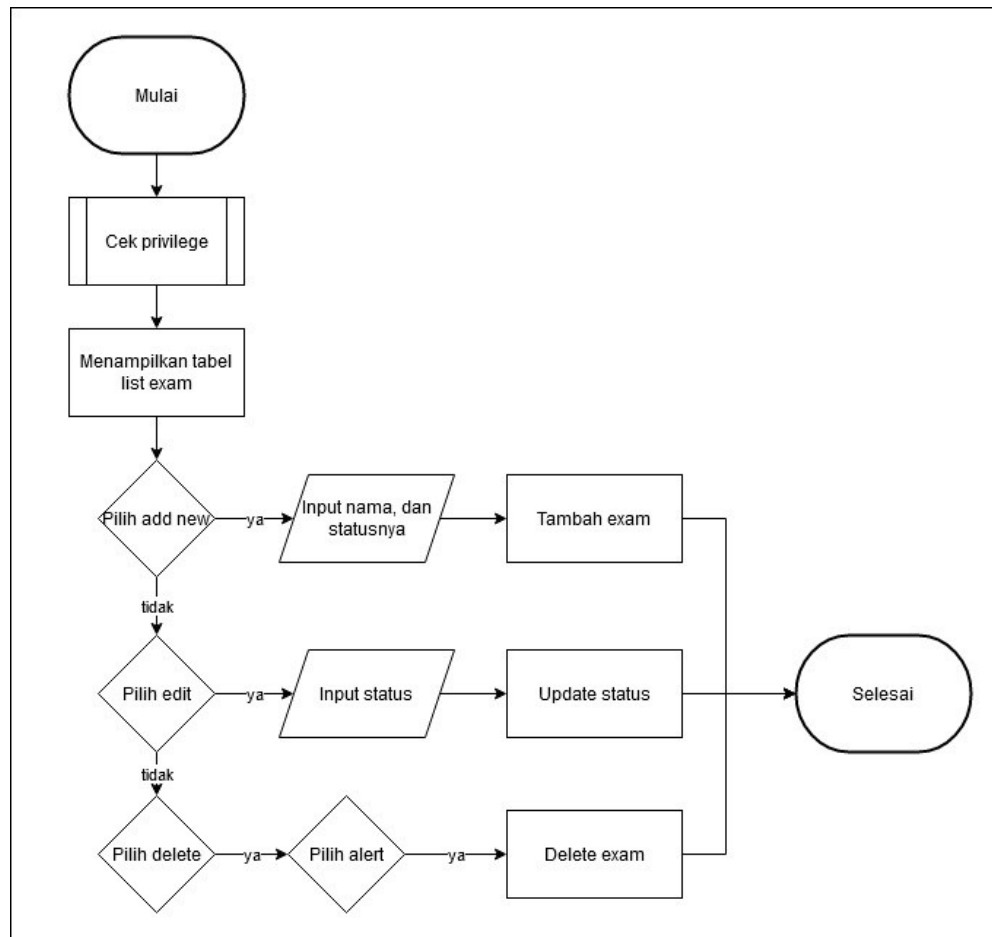
Gambar 3.11 Flowchart konfigurasi archived

Pada halaman CMS untuk monitor semua *request* kebutuhan, semua daftar *request* akan ditampilkan dalam bentuk tabel. Semua data akan diambil terlebih dahulu, dan dilanjutkan dengan mengambil *attachment* yang sesuai dengan masing-masing *request*. *Attachment* akan dapat diunduh dari *storage* jika menekan tombol *download*. Berikut adalah *flowchart* dari halaman untuk monitor *request* kebutuhan.



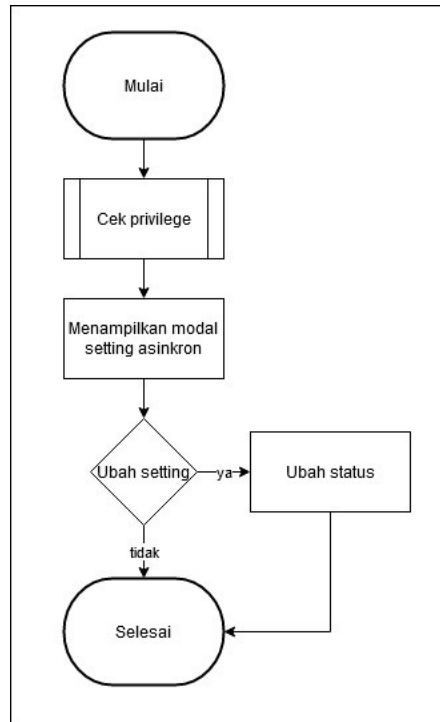
Gambar 3.12 Flowchart monitor request

Pada halaman CMS untuk konfigurasi *exam* diawali dengan menampilkan daftar tabel dari ujian-ujian. Pada masing-masing baris dapat melakukan *edit* dan *delete*. Penekanan tombol *edit* akan mengubah status dari *exam* yang ditekan, dan penekanan pada tombol *delete* akan menghapus *exam* dari daftar. Penambahan daftar juga dapat dilakukan dengan menekan tombol '*add*'. Berikut adalah *flowchart* dari halaman CMS untuk konfigurasi *exam*.



Gambar 3.13 Flowchart konfigurasi exam

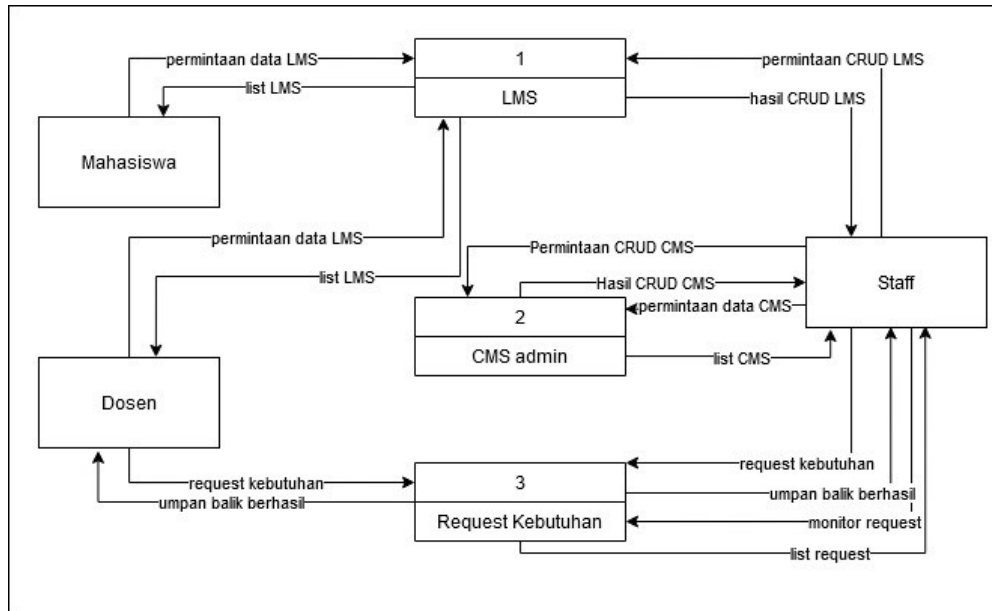
Pada halaman CMS untuk konfigurasi asinkron akan memunculkan modal berisi *form* yang menerima *input* untuk mengubah status dari asinkron sekarang. Jika *form* tersebut dikirim, maka akan mengubah status dari asinkron. Status tersebut akan berisi *on/off* untuk mengetahui statusnya. Berikut adalah *flowchart* dari halaman CMS untuk konfigurasi asinkron.



Gambar 3.14 Flowchart konfigurasi asinkron

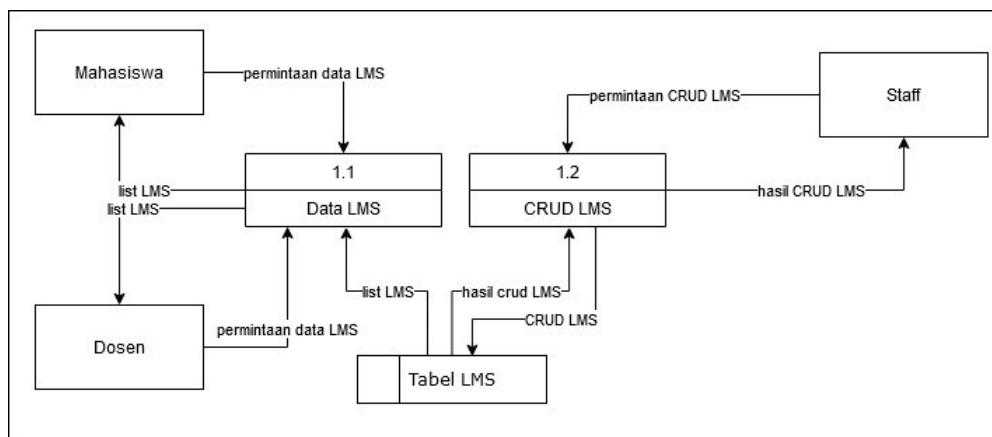
B. Diagram Aliran Data

Fitur baru dari sistem LCP yang dibuat oleh pelaksana kerja magang terdiri dari 3 subproses yaitu LMS, CMS untuk admin, dan *request* kebutuhan. Mahasiswa hanya dapat melihat LMS, dan dosen hanya dapat melihat LMS dan melakukan *request* kebutuhan. Sedangkan staff dari *learning technical development* akan dapat mengakses LMS, mengisi dan monitor *request* kebutuhan, dan mengatur CMS admin. Berikut adalah diagram aliran data level 0 dari fitur yang dikerjakan pelaksana kerja magang.



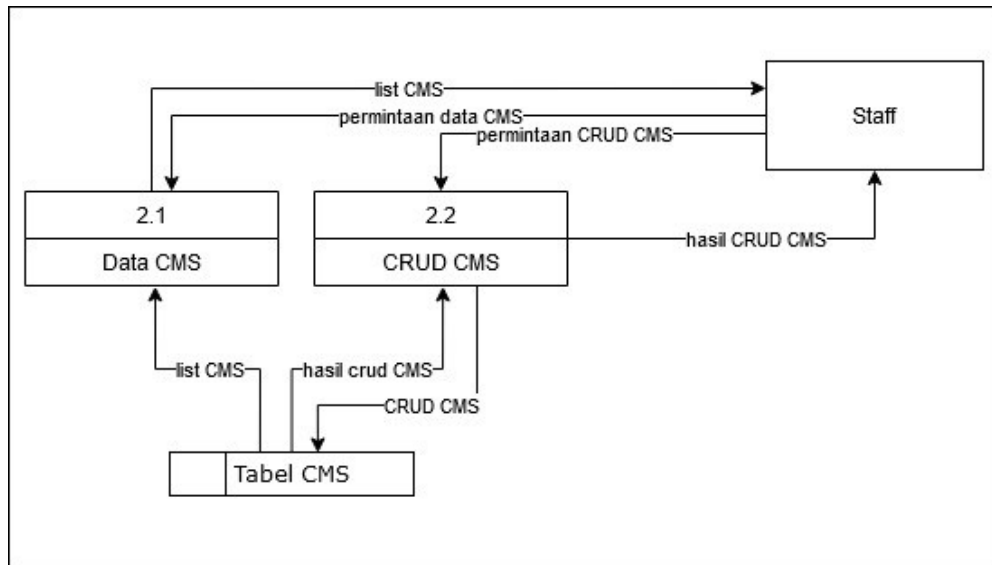
Gambar 3.15 DFD level 0 fitur LCP

Subsistem LMS akan berisi data LMS yang akan menampilkan berbagai macam LMS yang ada, dan CRUD (*Create Read Update Delete*) LMS yang akan mengatur daftar yang ada. CRUD dari LMS hanya dapat diakses oleh staff saja, dan data LMS dapat diakses oleh semua entitas. Berikut adalah DFD level 1 dari subsistem LMS.



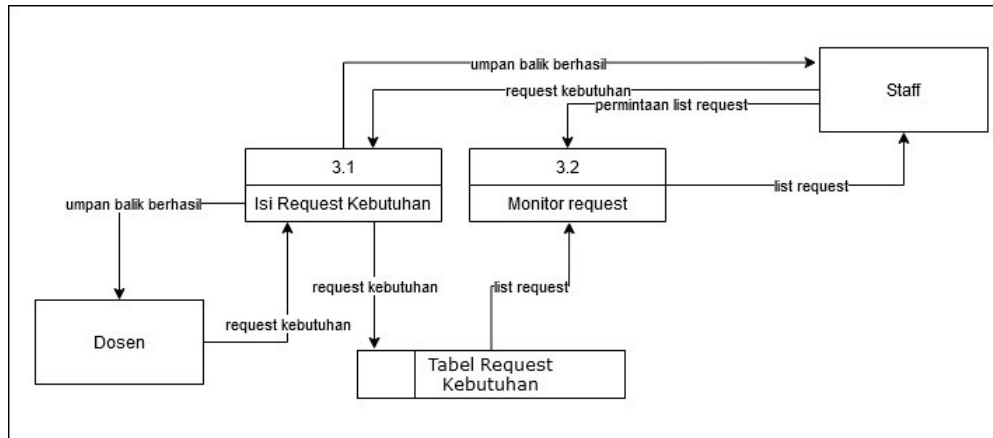
Gambar 3.16 DFD level 1 fitur LMS

Subsistem dari CMS admin akan menampilkan CMS-CMS yang sudah ada. Masing-masing isi dari CMS dapat dilihat oleh staff, dan juga ditambah, diubah, dan juga dihapus. Berikut adalah DFD level 1 dari subsistem CMS admin.



Gambar 3.17 DFD level 1 fitur CMS admin

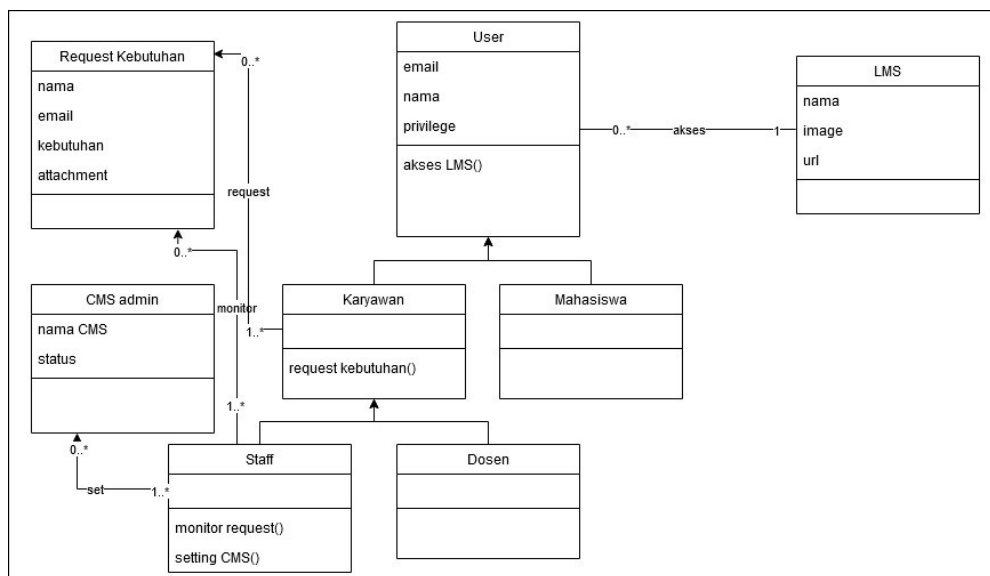
Subsistem dari *request* kebutuhan akan berisi form untuk isi *request* kebutuhan, dan halaman untuk monitor *request*. Halaman dari monitor request hanya dapat diakses oleh dosen dan staff. Sedangkan halaman untuk monitor *request* hanya dapat diakses oleh staff. Berikut adalah DFD level 1 dari fitur request kebutuhan.



Gambar 3.18 DFD level 1 fitur request kebutuhan

C. Class Diagram

Berikut adalah *class diagram* dari sistem LCP. Pada sistem LCP semua user akan dapat mengakses LMS untuk melihat daftar E-Learning yang tersedia. Halaman untuk *request* kebutuhan hanya dapat diakses dan diminta oleh karyawan saja yang meliputi dosen dan staff. Dan CMS untuk admin yang merupakan halaman untuk kendali hanya dapat diakses oleh staff/admin.



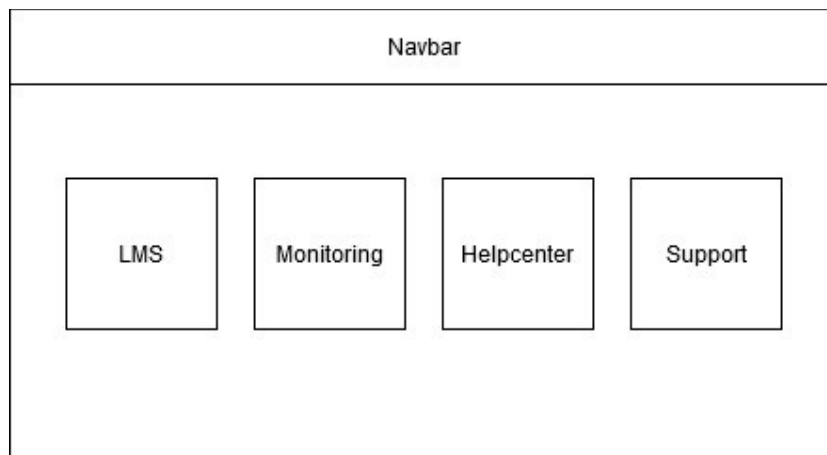
Gambar 3.19 Class diagram fitur yang dibuat

3.3.2. Desain Tampilan Antarmuka

Perancangan tampilan antarmuka dari masing-masing fitur dibuat oleh pelaksana kerja magang sesuai dengan keinginan dari *supervisor*. *Supervisor* akan memberikan bayangan mengenai tampilannya dan lalu dibuat rancangannya. Berikut adalah desain *mock up* dari halaman *learning center portal*, *request* kebutuhan, dan CMS admin.

A. Learning Center Portal

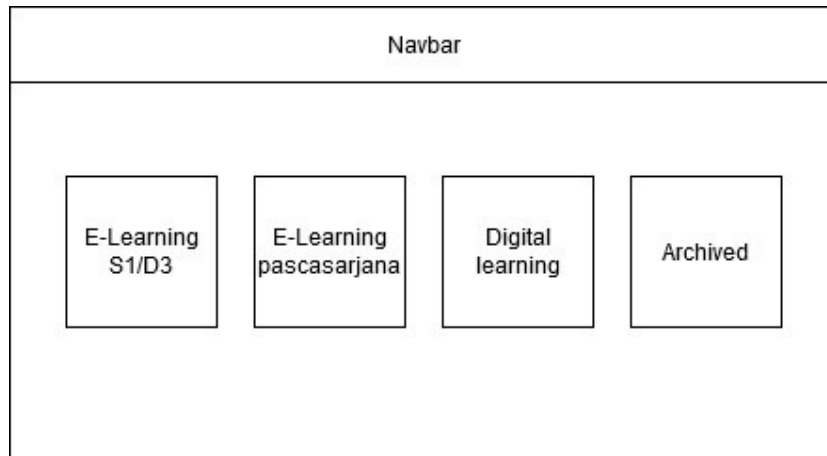
Perancangan halaman utama untuk LCP akan terdiri dari *navigation bar* (navbar) dan 4 tombol yaitu LMS, *monitoring*, *helpcenter*, dan *support*. Navbar akan berisi logo LCP, menu-menu yang ada, dan logo UMN untuk login. Keempat tombol yang ada akan pindah ke halamannya masing-masing jika ditekan. Berikut adalah rancangan tampilan dari menu utama LCP.



Gambar 3.20 Rancangan tampilan menu utama

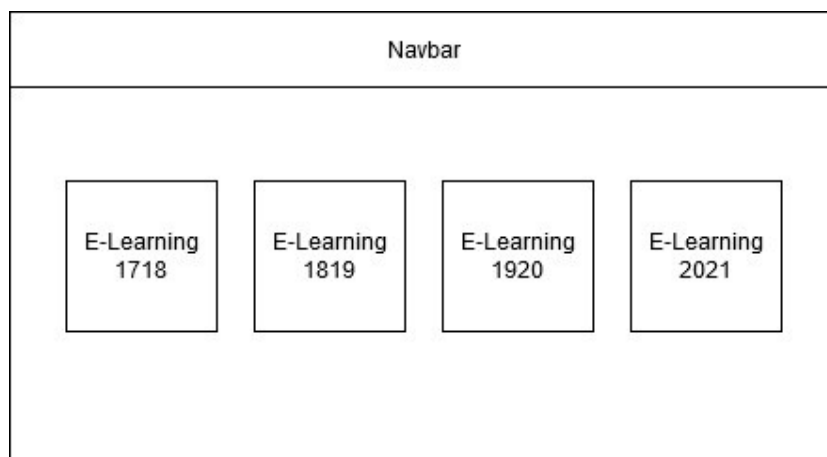
Perancangan tampilan dari menu LMS juga tidak jauh beda dengan tampilan menu utama dari LCP. Navbar masih berisikan logo LCP, dan 4 tombol yaitu E-

Learning S1/D3, E-Learning pascasarjana, *digital learning*, dan *archived*. Berikut adalah rancangan tampilan dari LMS.



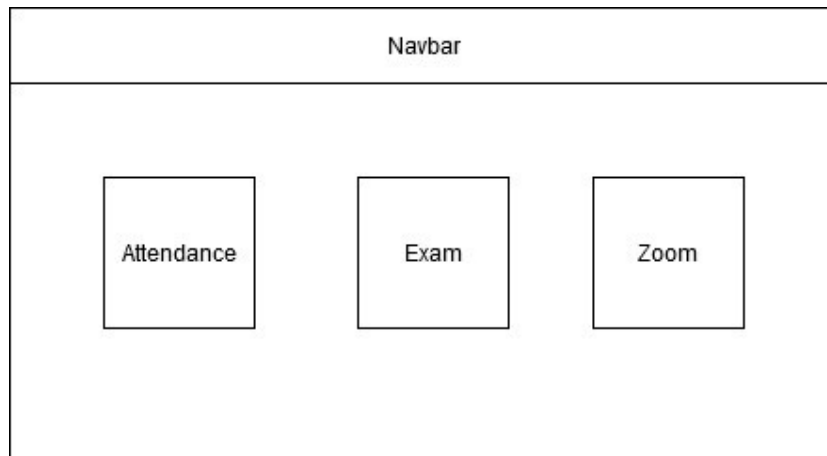
Gambar 3.21 Rancangan tampilan LMS

Perancangan tampilan dari E-Learning *archived* akan sama juga dengan halaman utama dan menu LMS yaitu 4 tombol sejajar. Tombol-tombol yang ada akan dibuat maksimal 4 untuk perbarisnya. Jika jumlah yang ada lebih dari 4, maka tombolnya akan berada dibawah barisnya. Berikut adalah rancangan tampilan dari halaman E-Learning *archived*.



Gambar 3.22 Rancangan tampilan archived

Perancangan tampilan dari menu *support* akan terdiri dari navbar dan 3 tombol yaitu *attendance*, *exam*, dan *zoom*. Ketiga tombol dapat ditekan dan akan memindahkan halaman ke halaman yang dituju. Berikut adalah rancangan tampilan dari halaman *support*.



Gambar 3.23 Rancangan tampilan menu support

B. Request Kebutuhan

Rancangan tampilan dari *request* kebutuhan akan berisi navbar dan *form*. *Form* akan memiliki judul di atas nya dan dilanjutkan dengan *form input* yang akan meminta masukan dari pengguna. Berikut adalah rancangan tampilan dari halaman *request* kebutuhan.

Navbar
<div style="border: 1px solid black; padding: 10px; margin: 0 auto; width: 60%;"> <p style="text-align: center;">Form Request Kebutuhan</p> <div style="margin-top: 10px;"> <input style="width: 90%; border: 1px solid black;" type="text"/> <input style="width: 90%; border: 1px solid black;" type="text"/> <input style="width: 90%; border: 1px solid black;" type="text"/> <input style="width: 90%; border: 1px solid black;" type="text"/> <input style="width: 90%; border: 1px solid black;" type="text"/> </div> </div>

Gambar 3.24 Rancangan tampilan menu support

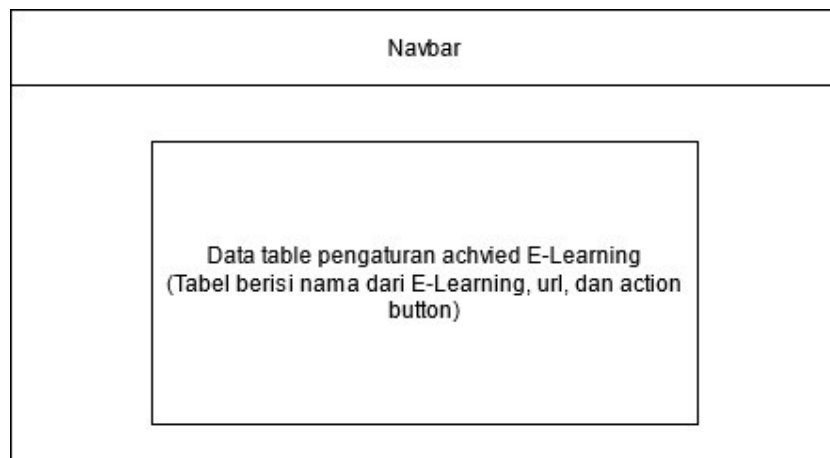
C. CMS Admin

Rancangan tampilan dari halaman menu CMS admin akan berisikan navbar dan *list* dari semua CMS yang sudah tersedia. Pembagian dari CMS untuk admin adalah per bagian. Nama menu akan dicetak bold, dan CMS nya tidak untuk membedakan. Berikut adalah rancangan tampilan dari halaman CMS untuk admin.

Navbar
<p>LMS</p> <p>Elearning</p> <p>Monitoring</p> <p>-</p> <p>Helpcenter</p> <p>-</p> <p>RMS</p> <p>Monitoring Request</p> <p>List Exam</p> <p>Setting - Asinkron</p>

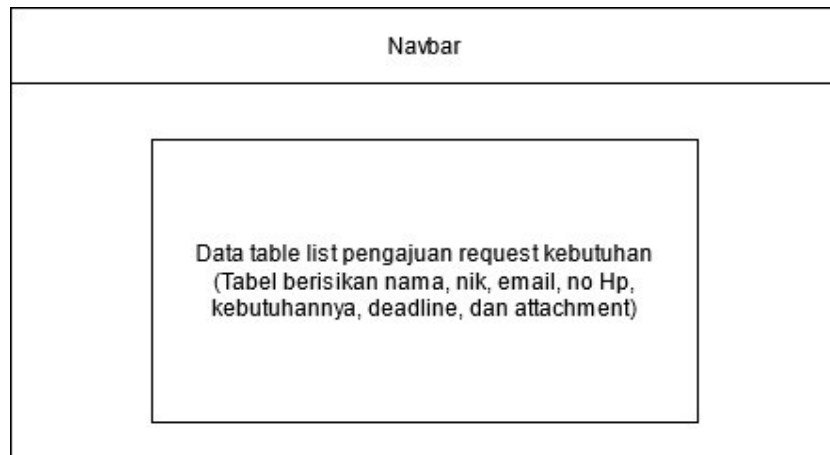
Gambar 3.25 Rancangan tampilan menu admin

Berikut adalah rancangan tampilan dari halaman untuk mengatur daftar E-Learning *archived*. Rancangan tampilannya akan berisikan navbar, dan tabel yang berisi daftar *archived* nya. Masing-masing baris akan ada tombol yang berguna untuk mengubah dan menghapus daftarnya. Tombol ‘*add*’ juga disediakan untuk menambah daftar *archived*.



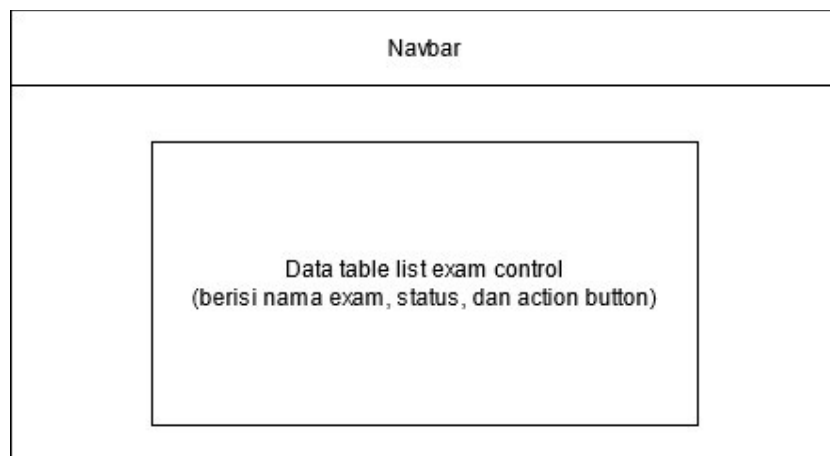
Gambar 3.26 Rancangan tampilan konfigurasi archived

Berikut adalah rancangan tampilan dari CMS untuk monitoring semua *request* yang ada. Rancangan tampilannya akan berisikan navbar, dan tabel yang berisi data. Tombol untuk *download* akan ada untuk mengunduh *attachment* yang ada atau tersedia.



Gambar 3.27 Rancangan tampilan monitor request

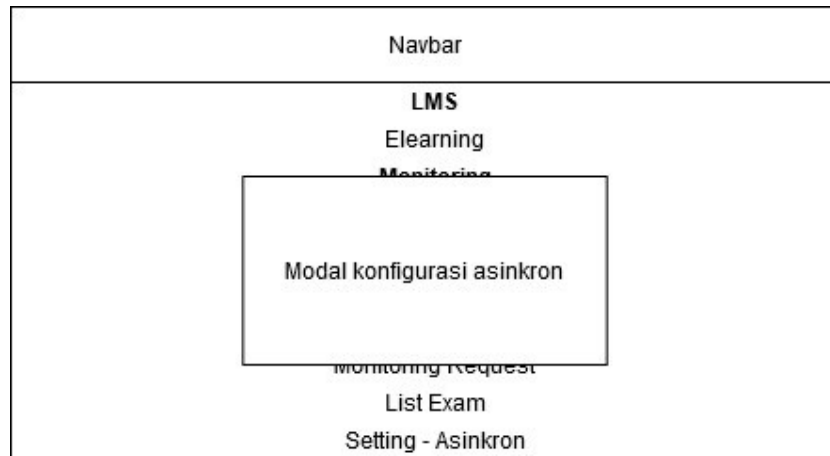
Berikut adalah rancangan tampilan dari CMS untuk *exam control*. Rancangan tampilannya akan berisikan navbar, dan tabel yang berisi data *exam*. Tombol untuk pengendalian akan ada di masing-masing daftar *exam*. Tombol tersebut adalah untuk *edit* dan *delete*.



Gambar 3.28 Rancangan tampilan exam control

Berikut adalah rancangan tampilan dari CMS untuk mengatur apakah asinkron aktif atau tidak aktif. Rancangan tampilannya akan masih di halaman awal

CMS admin, dan akan menampilkan *modal*. Modal akan berisikan *form select* untuk mengubah status.



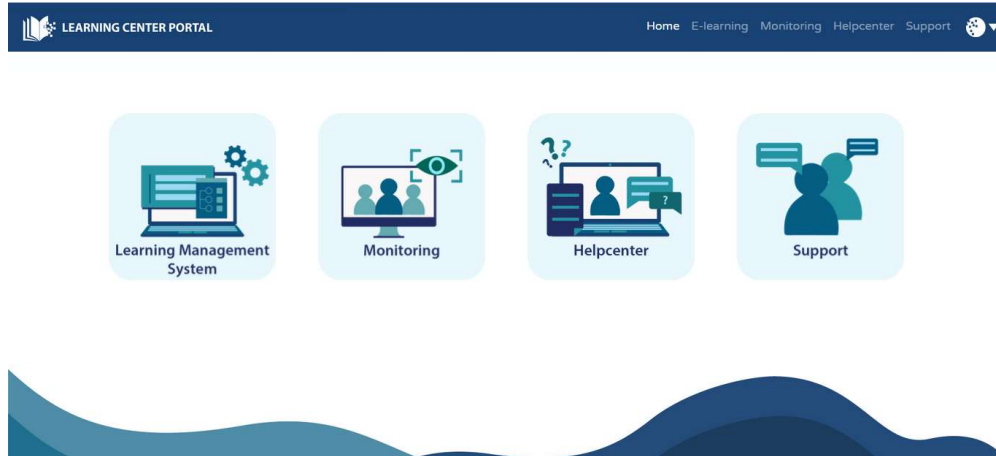
Gambar 3.29 Rancangan tampilan setting asinkron

3.3.3. Hasil Implementasi Halaman

Pengaturan tampilan dari halaman dibuat dengan menggunakan bootstrap dan juga css. Aset-aset yang ada dibuat dari UI/UX *designer* yang berada di bagian Learning Content Development. Berikut adalah hasil implementasi dari tampilan yang telah disebutkan pada perancangan aplikasi.

A. Learning Center Portal

Halaman menu awal dari Learning Center Portal dibuat sesuai dengan desain tampilan antarmuka yang diinginkan, yaitu memiliki sebaris 4 button. Urutannya juga disesuaikan dari LMS, monitoring, helpcenter, lalu support. Kemudian jika *user* memilih salah satu tombolnya, maka *user* akan mengunjungi halaman tersebut. Berikut adalah hasil implementasi halaman menu utama dari LCP.



Gambar 3.30 Implementasi halaman menu utama

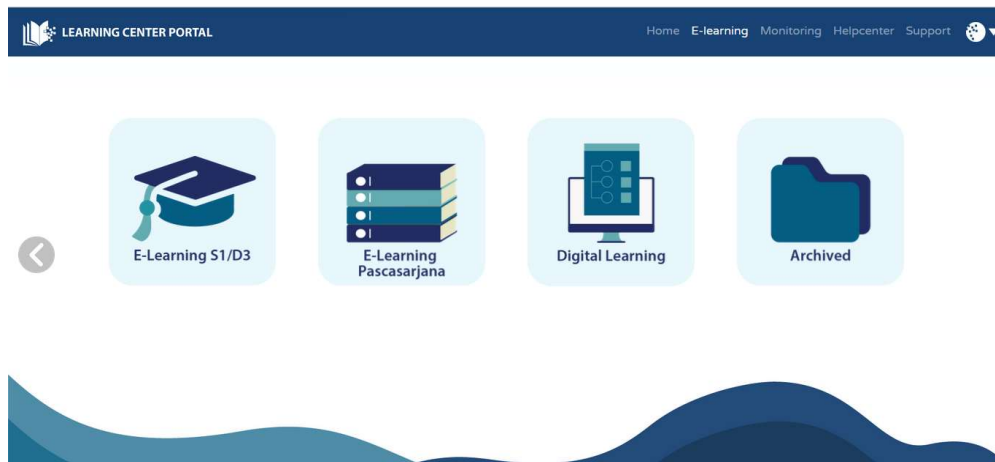
Berikut adalah potongan kode untuk menampilkan halaman menu utama dari LCP. Pengaturan tombolnya dilakukan dengan menggunakan kelas *row*, *no-gutters*, dan *clearfix* untuk mengatur agar masing-masing tombol dapat berada pada satu baris yang sama. Lalu ditambah dengan penggunaan kelas pembagian col yang membuat ukurannya menjadi responsif.

```
<div class="row no-gutters clearfix">
  <div class="col-sm-6 col-md-6 col-lg-3">
    <a href="elearning">
      
    </a>
  </div>
  <div class="col-sm-6 col-md-6 col-lg-3">
    <a href="monitoring/statistic/dosen">
      
    </a>
  </div>
  <div class="col-sm-6 col-md-6 col-lg-3">
    <a href="helpcenter#1">
      
    </a>
  </div>
  <div class="col-sm-6 col-md-6 col-lg-3">
    <a href="others">
      
    </a>
  </div>
</div>
```

Gambar 3.31 Potongan kode menampilkan tombol menu LCP

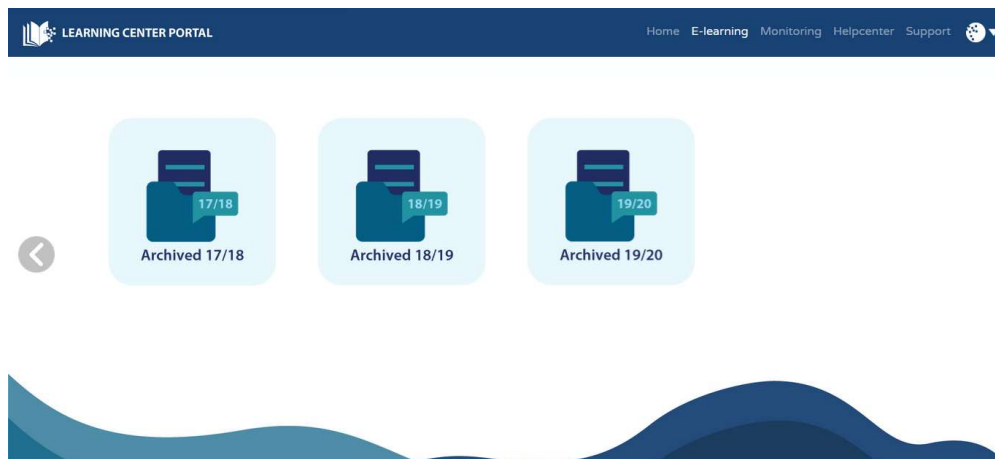
Berikut adalah halaman yang ditampilkan jika mengklik tombol LMS. Halaman LMS akan menampilkan 4 tombol yaitu E-Learning S1/D3, E-Learning

pascasarjana, digital learning, dan archived. Pada halaman LMS juga ditambahkan adanya tombol kembali yang berguna untuk kembali ke halaman sebelumnya yaitu halaman utama dari LCP.



Gambar 3.32 Implementasi halaman LMS

Berikut adalah halaman yang ditampilkan jika mengklik tombol *archived*. Halaman *archived* akan menampilkan daftar dari E-Learning yang pernah ada sebelumnya seperti elearning17/18, elearning 18/19 dan elearning 19/20. Isinya akan bisa bertambah seiring dengan berjalannya waktu.



Gambar 3.33 Implementasi halaman archived

Berikut adalah potongan kode yang digunakan untuk mengambil daftar *archived* yang disimpan pada database. Data diambil dari tabel 'el_archived' yang lalu dikirim ke *view* untuk ditampilkan. Variabel email dikirim juga agar *view* dapat mengetahui email dari yang sedang mengakses.

```
public function archived() {  
    $email = $this->email;  
  
    $data = DB::table('el_archived')->get();  
  
    return view('elearningArchived', compact('email', 'url', 'data'));
```

Gambar 3.34 Potongan kode ambil daftar archived

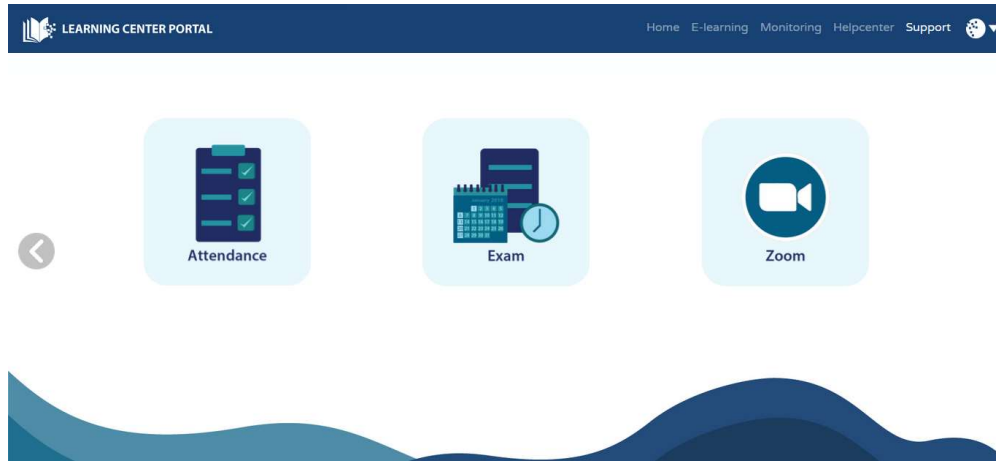
Berikut adalah potongan kode yang digunakan untuk menampilkan daftar *archived* yang berhasil didapat dari *controller*. Data ditampilkan dengan menggunakan *foreach* agar dapat menampilkan semua data yang didapat. Variabel url digunakan untuk perpindahan ke halamannya, dan variabel image untuk menampilkan gambar. Variabel nama dari data yang didapat juga digunakan pada id dari masing-masing tombolnya.

```
@foreach ($data as $item)  
  
    <div class="col-sm-6 col-md-6 col-lg-3">  
        <a href="{{ $item->url }}" target="_blank">  
            <img class="hvr-grow main-button" id="{{ $item->nama }}" src="{{ asset('uploads/' . $item->image) }}" />  
        </a>  
    </div>  
@endforeach
```

Gambar 3. 35 Potongan kode menampilkan daftar archived

Berikut adalah halaman yang ditampilkan pada halaman *support*. Halaman *support* dapat diakses dengan menekan tombol menu *support* yang ada pada

halaman utama LCP. Halaman ini menampilkan 3 tombol yaitu *attendance*, *exam*, dan *zoom*.



Gambar 3.36 Implementasi halaman support

B. Fitur Filter

Berikut adalah halaman yang dibuat untuk uji implementasi dari penggunaan fitur untuk memfilter. Akan ada *form* untuk *input* untuk pengguna, tombol untuk *submit*, dan tabel yang menampilkan semua data yang ada. Jika isi *input* penggunaanya kosong, maka akan menampilkan semua data.

LEARNING CENTER PORTAL

Home E-learning Monitoring Helpcenter Support

Host

Submit Query

ID	Nama Account	Kegiatan
1	3	Info Franchise & Business Concept Expo 2020
2	6	INDONESIA ENVIROTECH EXPO & FORUM 2020
3	1	INDONESIA STEEL BUILDING & METAL STRUCTURE EXPO 2020
4	12	6TH CHEMICAL EXPO 2020
5	2	IBME (Indonesia Building Mechanical Electrical Expo), LIFT & ESCALATOR, HVAC & ENERGY EXPO
6	11	Indonesia Smart Security Fire & Rescue Expo 2020
7	4	INDOSOLAR Expo & Forum 2020
8	10	INDONESIA SMART BUILDING SMARTCITY WEEK 2020

Gambar 3.37 Implementasi filter menampilkan semua data

Berikut adalah hasil implementasi filter saat sudah memilih apa yang ingin ditampilkan. Semua angka yang ada di form akan diambil datanya dan ditampilkan. Data yang ditampilkan adalah nama akun yang dicari dan datanya.

LEARNING CENTER PORTAL

Home E-learning Monitoring Helpcenter Support

Host

3,6,10

Submit Query

ID	Nama Account	Kegiatan
1	3	Info Franchise & Business Concept Expo 2020
28	3	ELENEX INDONESIA
2	6	INDONESIA ENVIROTECH EXPO & FORUM 2020
25	6	Indonesia International Book Fair
38	6	Kustomfest 2020
8	10	INDONESIA SMART BUILDING SMARTCITY WEEK 2020
21	10	Retail & Solutions Expo Indonesia – RSEI
31	10	PAMERAN PALEMBANG ITT 2020

Gambar 3.38 Implementasi filter

Berikut adalah potongan kode dari fitur filter yang mengambil masukan dari pengguna dan dimasukkan pada variabel req. Hasil masukan tersebut dilanjutkan dengan memroses datanya dimasukkan ke variabel int_array dengan menghilangkan koma, tanda penghubung, dan spasi. Dilanjutkan dengan pengecekan apabila terdapat tanda penghubung pada masukan pengguna. Jika didapat ternyata ada tanda penghubung, maka angka sebelum tanda tersebut dan setelahnya akan ditambahkan pada int_array.

```
public function filter(Request $req) {  
    $email = $this->email;  
    $string = $req->input('host');  
  
    $temp = $string;  
    $int_array = array_map("intval", $this->multiexplode(array(",","-"," "), $temp));  
  
    $length = count($int_array);  
    $firstNum;  
    $lastNum;  
    $stringLength = strlen($string);  
  
    if(strpos($string, '-') !== false){  
        for($i=0; $i < $stringLength; $i++){  
            if(is_numeric($string[$i])){  
                continue;  
            }  
            else if($string[$i] == '-'){  
                $firstNum = $int_array[$i/2 -1];  
                if($i % 2 == 0) $lastNum = $int_array[floor($i/2)];  
                else $lastNum = $int_array[floor($i/2 + 1)];  
  
                for($j = $firstNum+1; $j < $lastNum; $j++){  
                    $string .= ",$j";  
                }  
            }  
        }  
    }  
    $int_array = array_map("intval", $this->multiexplode(array(",","-"," "), $string));  
}
```

Gambar 3.39 Potongan kode fungsi filter

Setelah mendapat angka-angka yang dibutuhkan untuk ditampilkan, Langkah selanjutnya adalah dengan mengambil data yang ada dari basis data. Dilakukan pengecekan terlebih dahulu yang apabila tidak masukan dari pengguna, maka akan menampilkan semuanya. Lalu jika inputnya hanya satu, langsung menampilkan

data tersebut saja, dan apabila inputnya banyak, maka akan dipanggil datanya dengan menggunakan fungsi `whereIn`. Berikut adalah potongan kode dari mengambil data sesuai dari masukkan pengguna.

```
if($string == null)
    $data = DB::table('test_zoom')->get();
elseif($length == 1)
    $data = DB::table('test_zoom') ->where('accountid', $int_array[0])
    ->get();
elseif($length > 1){
    $data = DB::table('test_zoom') ->whereIn('accountid', $int_array)
    ->orderBy('accountid', 'asc')
    // ->groupBy('accountid')
    ->get();
}

return view('others.filter.index', compact('email', 'data'));
```

Gambar 3.40 Potongan kode mengambil data filter

C. Request Kebutuhan

Berikut adalah hasil implementasi halaman dari halaman untuk *request* kebutuhan. Halaman akan menampilkan *form* yang akan meminta nama, NIK, email, nomor *handphone*, kebutuhan, *attachment*, dan batas tanggal untuk *deadline*. Untuk tanggal *deadline* dengan minimal 2 hari kerja. *Form* dari *request* kebutuhan diberikan border untuk menjadi pembeda dengan latar belakang.

LEARNING CENTER PORTAL

Home E-learning Monitoring Helpcenter Support

Request Kebutuhan

Nama

NIK
00000012345

Email
name@example.com

No Handphone

Kebutuhan

Gambar 3.41 Implementasi request kebutuhan

Berikut adalah tampilan implementasi dari menampilkan tanggal. Tanggal ini bisa dipilih dengan minimal waktu 2 hari kerja. Tampilan *date picker* menggunakan *jquery ui* agar dapat dibuka juga di *browser* Safari.

LEARNING CENTER PORTAL

Home E-learning Monitoring Helpcenter Support

No Handphone

Kebutuhan

November 2020

Su	Mo	Tu	We	Th	Fr	Sa
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

Submit

Gambar 3.42 Implementasi penampilan tanggal

Berikut adalah potongan kode untuk menampilkan batas akhir dari *datepicker* yang memilih minimal 2 hari waktu kerja untuk dipilih. Yang dilakukan pertama adalah mendapatkan tanggal dari hari tersebut, dilanjutkan dengan menentukan hari kerja nya adalah 2 hari. Setelahnya adalah dengan mengecek apakah hari besok atau

lusanya merupakan akhir pekan, yang jika salah satunya merupakan akhir pekan maka hari kerjanya ditambahkan 2 lagi.

```
public function index() {
    $email = $this->email;
    $dayOne = date('Y-m-d', strtotime(date('Y-m-d').' + '.1.' days'));
    $dayTwo = date('Y-m-d', strtotime(date('Y-m-d').' + '.2.' days'));
    $dayOneisWeekend = date('N', strtotime($dayOne)) >= 6;
    $dayTwoisWeekend = date('N', strtotime($dayTwo)) >= 6;
    $minDate = 2;

    if($dayOneisWeekend){
        $minDate+=2;
    }
    else if($dayTwoisWeekend){
        $minDate+=2;
    }

    return view('rms.ticket.index', compact('email', 'minDate'));
}

function isWeekend($date) {
    return (date('N', strtotime($date)) >= 6);
}
```

Gambar 3.43 Potongan kode atur minimal 2 hari

D. CMS Tukar PIC

Pada CMS tukar PIC, terdapat halaman untuk ganti email, *form* tukar PIC nya, dan halaman yang menampilkan *request* untuk PIC nya. Berikut adalah implementasi halaman untuk ganti email dari CMS untuk tukar PIC. *Form* akan meminta user dan email yang ingin diganti.

LEARNING CENTER PORTAL

Home E-learning Monitoring Helpcenter Support

Ganti Email

User yang mengajukan ganti email

BIA3

Email

michael.sihotang@student.umn.ac.id

Submit

Gambar 3.44 Implementasi ganti email

Berikut adalah potongan kode yang digunakan untuk mengganti alamat email. *Controller* akan menerima masukan dari pengguna, yang dilanjutkan dengan mengubah email dari PIC. Ini digunakan apabila terjadi pergantian PIC dari masing-masing bagian.

```
public function validasi(Request $req) {  
    $user = $req->user;  
    $email = $req->email;  
  
    DB::update('update cms_user set email = ? where user = ?', [$email, $user]);  
  
    return redirect('/others/cms');  
}
```

Gambar 3.45 Potongan kode ganti email PIC

Berikut adalah implementasi halaman untuk penukaran PIC. Form akan meminta pengguna dan tanggal yang ingin diajukan untuk ditukar. Saat tombol *submit* ditekan, akan mengirimkan pengajuan ke pengguna tersebut untuk disetujui atau tidak.

Gambar 3.46 Implementasi request tukar PIC

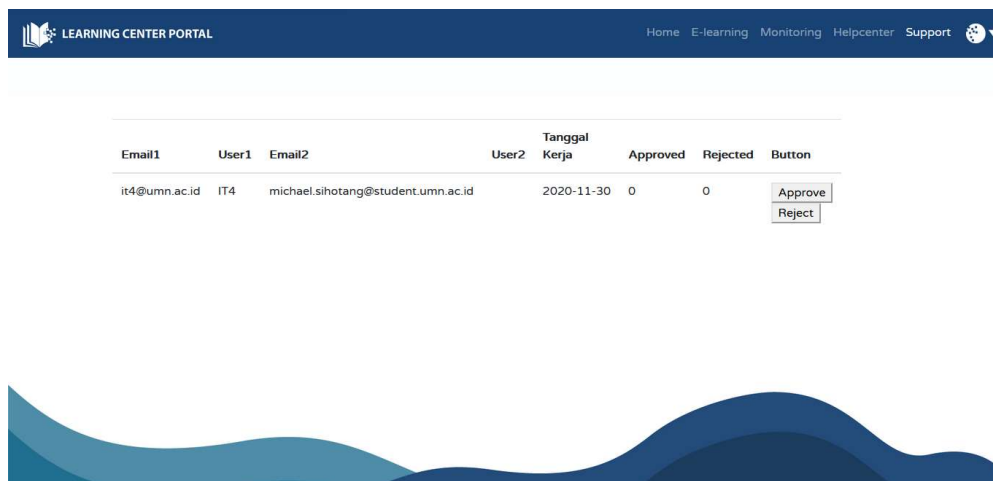
Berikut adalah potongan kode untuk menyimpan pengajuan tukar PIC. Yang dilakukan pertama adalah mendapatkan email dari yang diajukan sesuai dengan bagiannya. Dilanjutkan dengan menyimpan semua data yang ada ke basis data dan kembali ke halaman awal.

```
public function submits(Request $request) {
    $data = DB::table('cms_user')->where('user', $request->user2)
                                         ->get();
    foreach($data as $d){
        $email2 = $d->email;
    }

    DB::table('cms_tukar')->insert([
        'user1' => $request->user1,
        'user2' => $request->user2,
        'email1' => $this->email,
        'email2' => $request->email2,
        'tanggal' => $request->tanggal,
        'approved' => 0
    ]);
    return redirect('/others/cms');
}
```

Gambar 3.47 Potongan kode pengajuan tukar PIC

Berikut adalah halaman implementasi halaman untuk menampilkan *request* yang didapat. Halaman akan menampilkan daftar pengguna yang telah mengajukan untuk bertukar PIC. Pengguna yang diminta tukar dapat menyetujui atau menolak untuk bertukar pada hari yang diminta.



Email1	User1	Email2	User2	Tanggal Kerja	Approved	Rejected	Button
it4@umn.ac.id	IT4	michael.sihotang@student.umn.ac.id		2020-11-30	0	0	<input type="button" value="Approve"/> <input type="button" value="Reject"/>

Gambar 3.48 Implementasi request ditampilkan

Berikut adalah potongan kode untuk menyetujui dan menolak diajukannya tukar PIC. *Controller* akan menerima id dari pengajuan tersebut, dan mengupdate statusnya sesuai dari tombol yang ditekan pengguna. Setelahnya adalah dilanjutkan dengan Kembali ke halaman awal.

```

public function approved(Request $req) {
    $id = $req->id;

    DB::update('update cms_tuker set approved = 1 where id = ?', [$id]);

    return redirect('/others/cms');
}

public function rejected(Request $req) {
    $id = $req->id;

    DB::update('update cms_tuker set rejected = 1 where id = ?', [$id]);

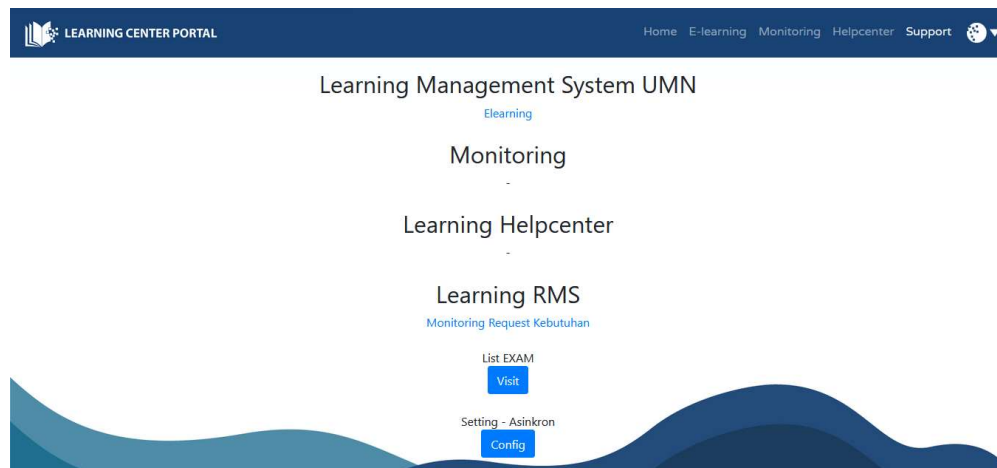
    return redirect('/others/cms');
}

```

Gambar 3.49 Potongan kode setuju dan tolak tukar PIC

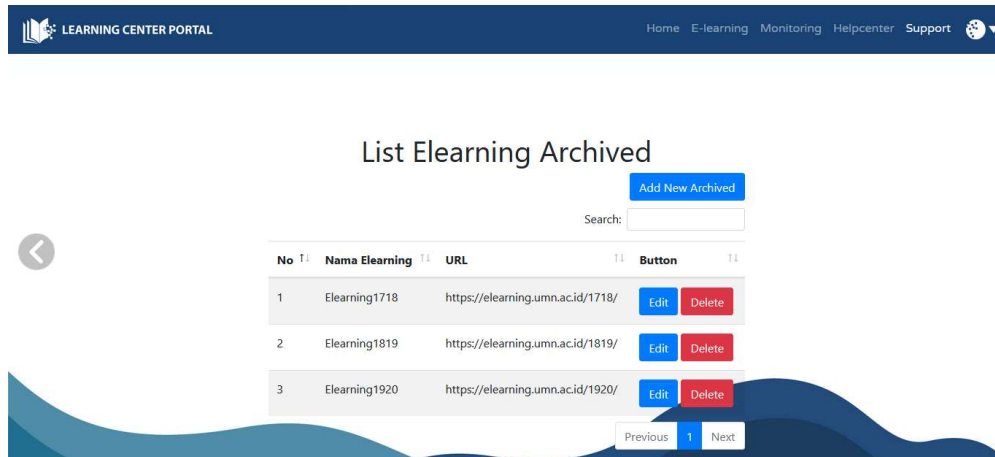
E. CMS Admin

Berikut adalah halaman implementasi untuk menampilkan daftar CMS untuk yang sudah tersedia. Halaman akan menampilkan daftar CMS yang dapat dipakai. Pada saat ini yang tersedia adalah untuk mengatur daftar *archived* E-Learning, monitoring *request* kebutuhan, pengaturan *exam*, dan pengaturan asinkron.



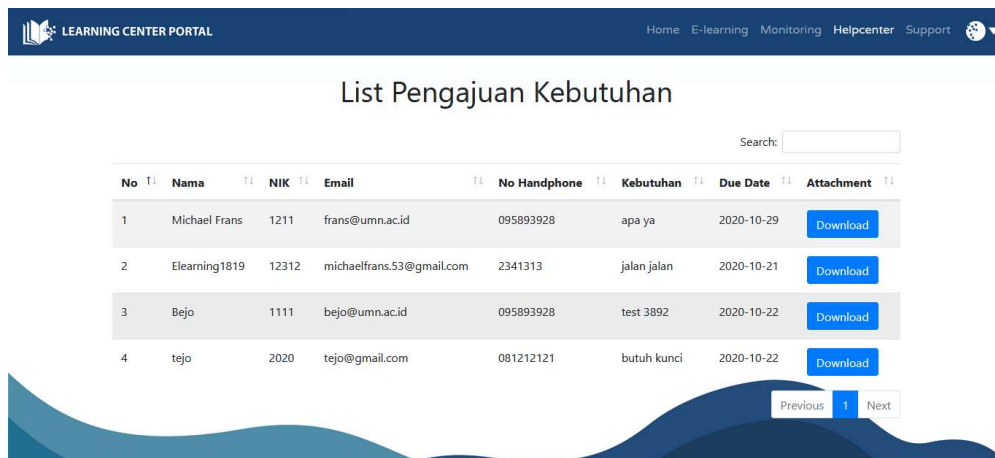
Gambar 3.50 Implementasi main menu CMS admin

Berikut adalah halaman implementasi untuk CMS E-Learning *archived*. Tombol *add* yang digunakan untuk tambah daftar E-Learning berada diatas tabel. Tabel akan berisikan nama elarning, url nya, dan tombol *edit* untuk mengubah dan tombol *delete* untuk menghapus.



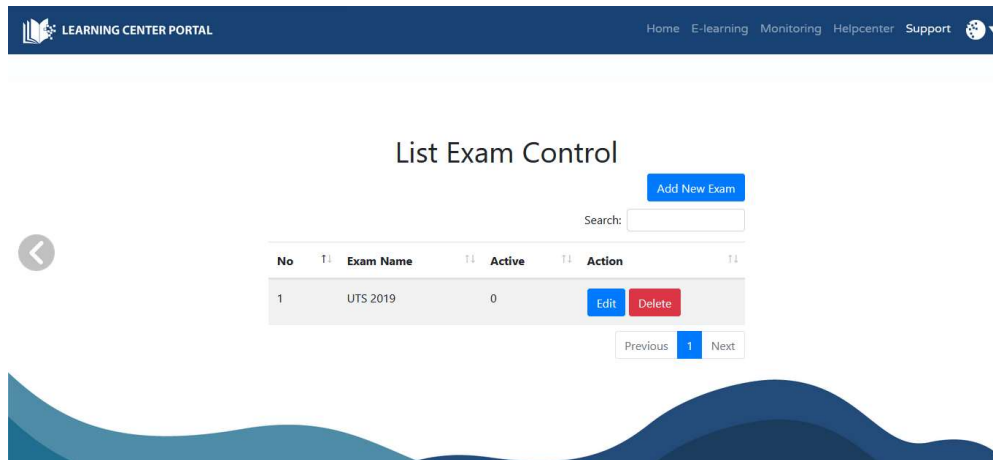
Gambar 3.51 Implementasi cms archived LMS

Berikut adalah hasil implementasi halaman untuk memantau hasil *request-request* yang ada dari pengajuan kebutuhan. Halaman akan menampilkan tabel yang berisi nama, NIK, email, nomor handphone, kebutuhannya, batas tanggal, dan tombol *download* untuk mengunduh *attachment*. Penampilan data ini menggunakan *data table* agar dapat melakukan pencarian dan juga pengurutan.



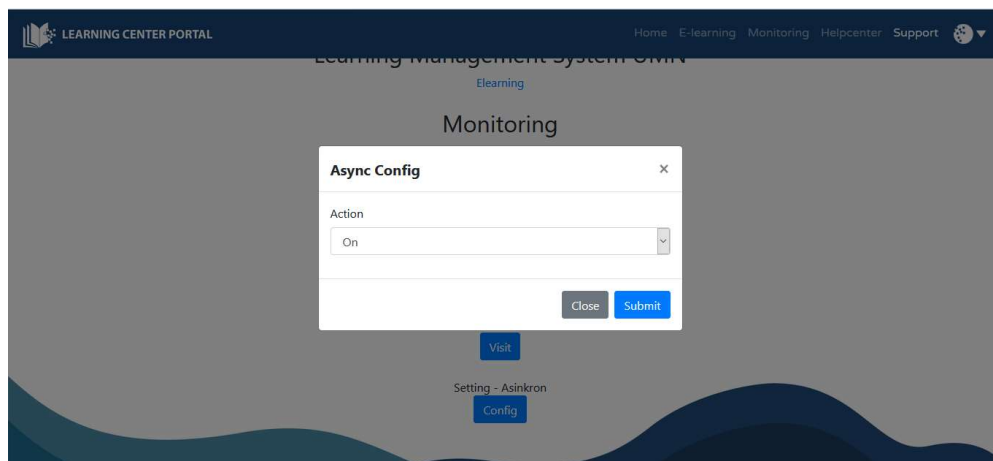
Gambar 3.52 Implementasi monitoring request kebutuhan

Berikut adalah implementasi halaman untuk mengatur *exam*. Halaman ini akan ada tabel yang berisi nama *exam*, status aktif, dan tombol *edit* dan *delete*. Tombol *add* untuk menambah daftar *exam* akan ada diatas tabel.



Gambar 3.53 Implementasi CMS exam

Berikut adalah implementasi halaman untuk mengatur apakah pengajuan asinkron aktif atau tidak. Tampilan akan berbentuk *modal* yang berisikan *form* yang meminta *input* untuk mengubah statusnya. Jika tombol *submit* ditekan, maka status dari pengajuan asinkron akan berubah.



Gambar 3.54 Implementasi konfigurasi asinkron

3.3.2. Kendala yang Ditemukan

Berikut adalah kendala-kendala yang ditemukan selama masa pelaksanaan kerja magang.

1. Adanya perubahan *requirement* yang dibutuhkan dalam pengembangan web.
2. Tidak diberikannya contoh data *dummy* dikarenakan banyaknya data-data kredensial.

3.3.3. Solusi atas Kendala yang Ditemukan

Berikut adalah solusi dari kendala-kendala yang telah ditemukan.

1. Menyesuaikan website yang telah dibuat dengan perubahan *requirement* tersebut.
2. Memberikan *comment* pada baris kode yang error karena tidak mendapatkan tabel yang dibutuhkan.